

JASPERREPORTS - REPORT STYLES

http://www.tutorialspoint.com/jasper_reports/jasper_report_styles.htm

Copyright © tutorialspoint.com

JasperReports has a feature `<style>` which helps controls text properties in a report template. This element is a collection of style settings declared at the report level. Properties like foreground color, background color, whether the font is bold, italic, or normal, the font size, a border for the font, and many other attributes are controlled by `<style>` element. Styles can extend other styles, and add to, or override properties of the parent style.

Style Properties

A `<style>` element has many attributes. Some of the most commonly used are listed in the table below:

Attribute	Description
name	Is mandatory. It must be unique because it references the corresponding report style throughout the report
isDefault	Indicates whether this style is the document's default style.
style	A reference to the parent style
mode	Specifies the element's transparency. Possible values are <i>Opaque</i> and <i>Transparent</i> .
forecolor	The foreground color of object.
backcolor	The background color of object.
fill	Determines the fill pattern used to fill the object. At the moment the single value allowed is <i>Solid</i> .
radius	Specifies the radius of the rectangle's corner arc.
scaleImage	Specifies the scale for images only. Possible values: <i>Clip</i> , <i>FillFrame</i> , <i>RetainShape</i> , <i>RealHeight</i> , <i>RealSize</i> .
hAlign	Specifies the horizontal alignment. Possible values: <i>Left</i> , <i>Center</i> , <i>Right</i> , <i>Justified</i> .
vAlign	Specifies the vertical alignment. Possible values: <i>Top</i> , <i>Middle</i> , <i>Bottom</i> .
rotation	Specifies the element's rotation. Possible values: <i>None</i> , <i>Left</i> , <i>Right</i> , <i>UpsideDown</i> .
lineSpacing	Specifies the line spacing between lines of text. Possible values: <i>Single</i> , <i>1_1_2</i> , <i>Double</i> .
markup	Specifies the markup style for styled texts
fontName	Specifies the font name.
fontSize	Specifies the font size.
isBold	Indicates if the font style is bold
isItalic	Indicates if the font style is italic.
isUnderline	Indicates if the font style is underline

isStrikeThrough	Indicates if the font style is strikethrough.
pdfFontName	Specifies the related PDF font name.
pdfEncoding	Specifies the character encoding for the PDF output format
isPdfEmbedded	Indicates if the PDF font is embedded.
pattern	Specifies the format pattern for formatted texts.
isBlankWhenNull	Indicates if a white space should be shown if the text is not present.

Conditional Styles

In some situations, a style should be applied only when certain condition is met (for example, to alternate adjacent row colors in a report detail section). This can be achieved using conditional styles.

A conditional style has two elements:

- a Boolean condition expression
- a style

The style is used only if the condition evaluates to *true*.

Applying Styles to Report Elements

Any type of report element can reference a report style definition using the style attribute. Hence all the style properties declared by the style definition that are applicable to the current element will be inherited. To override the inherited values, style properties specified at the report element level can be used.

Style Templates

We can make a set of reports with a common look by defining the style at a common place. This common style template can then be referenced by the report templates. A style template is an XML file that contains one or more style definitions. Style template files use by convention the ***.jrtx** extension, but this is not mandatory. A style template contains following elements:

- `<jasperTemplate>`: This is the root element of a style template file.
- `<template>`: This element is used to include references to other template files. The contents of this element are interpreted as the location of the referred template file.
- `<style>`: This element is identical to the element with the same name from report design templates (JRXML files), with the exception that a style in a style template cannot contain conditional styles. This limitation is caused by the fact that conditional styles involve report expressions, and expressions can only be interpreted in the context of a single report definition.

References to style templates are included in JRXML reports as `<template>` elements. The style templates are loaded at report fill time, and style name references are resolved once all the templates have been loaded. When loading style templates and resolving style names to styles, a tree/graph of style templates is created, the top of the tree being the set of styles defined in the report. On this tree, style name references are resolved to the last style that matches the name in a depth-first traversal.

Example

Let's try out the conditional styles and style templates. Let's add the `<style>` element **alternateStyle** to our existing report template (Chapter [Report designs](#)). Based on the condition, font color changes to blue for even count. We have also included a style template "styles.jrtx". The revised report template (jasper_report_template.jrxml) is as follows. Save it to C:\tools\jasperreports-5.0.1\test directory:

```
<?xml version="1.0"?>
<!DOCTYPE jasperReport PUBLIC
"//JasperReports//DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">

<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="jasper_report_template" pageWidth="595"
pageHeight="842" columnWidth="515"
leftMargin="40" rightMargin="40" topMargin="50" bottomMargin="50">
  <template>"styles.jrtx"</template>
  <style name="alternateStyle" fontName="Arial" forecolor="red">
    <conditionalStyle>
      <conditionExpression>
        <![CDATA[new Boolean($V{countNumber}.intValue() % 2 == 0)]]>
      </conditionExpression>
      <style forecolor="blue" isBold="true"/>
    </conditionalStyle>
  </style>
  <parameter name="ReportTitle" />
  <parameter name="Author" />

  <queryString>
    <![CDATA[]]>
  </queryString>

  <field name="country" >
    <fieldDescription><![CDATA[country]]></fieldDescription>
  </field>

  <field name="name" >
    <fieldDescription><![CDATA[name]]></fieldDescription>
  </field>

  <variable name="countNumber" >
    <variableExpression><![CDATA[Boolean.TRUE]]></variableExpression>
  </variable>

  <title>
    <band height="70">
      <line>
        <reportElement x="0" y="0" width="515"
          height="1"/>
      </line>
      <textField isBlankWhenNull="true" bookmarkLevel="1">
        <reportElement x="0" y="10" width="515"
          height="30"/>
        <textElement textAlignment="Center">
          <font size="22"/>
        </textElement>
        <textFieldExpression >
          <![CDATA[$P{ReportTitle}]]>
        </textFieldExpression>
        <anchorNameExpression><![CDATA["Title"]]>
        </anchorNameExpression>
      </textField>
      <textField isBlankWhenNull="true">
        <reportElement x="0" y="40" width="515" height="20"/>
        <textElement textAlignment="Center">
          <font size="10"/>
        </textElement>
        <textFieldExpression >
          <![CDATA[$P{Author}]]>
        </textFieldExpression>
      </textField>
    </band>
  </title>
</jasperReport>
```

```

        </textFieldExpression>
    </textField>
</band>
</title>

<columnHeader>
    <band height="23">
        <staticText>
            <reportElement mode="Opaque" x="0" y="3"
                width="535" height="15"
                backcolor="#70A9A9" />
            <box>
                <bottomPen lineWidth="1.0"
                    lineColor="#CCCCCC" />
            </box>
            <textElement />
            <text><![CDATA[]]>
            </text>
        </staticText>
        <staticText>
            <reportElement x="414" y="3" width="121"
                height="15" />
            <textElement textAlignment="Center"
                verticalAlignment="Middle">
                <font isBold="true" />
            </textElement>
            <text><![CDATA[Country]]></text>
        </staticText>
        <staticText>
            <reportElement x="0" y="3" width="136"
                height="15" />
            <textElement textAlignment="Center"
                verticalAlignment="Middle">
                <font isBold="true" />
            </textElement>
            <text><![CDATA[Name]]></text>
        </staticText>
    </band>
</columnHeader>

<detail>
    <band height="16">
        <staticText>
            <reportElement mode="Opaque" x="0" y="0"
                width="535" height="14"
                backcolor="#E5ECF9" />
            <box>
                <bottomPen lineWidth="0.25"
                    lineColor="#CCCCCC" />
            </box>
            <textElement />
            <text><![CDATA[]]>
            </text>
        </staticText>
        <textField>
            <reportElement style="alternateStyle"
                x="414" y="0" width="121" height="15" />
            <textElement textAlignment="Center"
                verticalAlignment="Middle">
                <font size="9" />
            </textElement>
            <textFieldExpression >
                <![CDATA[${F{country}}]>
            </textFieldExpression>
        </textField>
        <textField>
            <reportElement x="0" y="0" width="136"
                height="15" style="Strong"/>
            <textElement textAlignment="Center"
                verticalAlignment="Middle" />
            <textFieldExpression >
                <![CDATA[${F{name}}]>
            </textFieldExpression>
        </textField>
    </band>
</detail>

```

```

        </textField>
    </band>
</detail>
</jasperReport>

```

The contents of style template **styles.jrtx** are as follows. Save it to C:\tools\jasperreports-5.0.1\test directory.

```

<?xml version="1.0"?>
<!DOCTYPE jasperTemplate
    PUBLIC "-//JasperReports//DTD Template//EN"
    "http://jasperreports.sourceforge.net/dtds/jasptemplate.dtd">

<jasperTemplate>
    <style name="Strong" isBold="true"
        pdfFontName="Helvetica-Bold" bgcolor="lightGray"
        forecolor="green"/>
</jasperTemplate>

```

The java codes for report filling remain unchanged. The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\JasperReportFill.java** are as below.

```

package com.tutorialspoint;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;

public class JasperReportFill {
    @SuppressWarnings("unchecked")
    public static void main(String[] args) {
        String sourceFileName =
            "C://tools/jasperreports-5.0.1/test/jasper_report_template.jasper";

        DataBeanList DataBeanList = new DataBeanList();
        ArrayList<DataBean> dataList = DataBeanList.getDataBeanList();

        JRBeanCollectionDataSource beanColDataSource =
            new JRBeanCollectionDataSource(dataList);

        Map parameters = new HashMap();
        /**
         * Passing ReportTitle and Author as parameters
         */
        parameters.put("ReportTitle", "List of Contacts");
        parameters.put("Author", "Prepared By Manisha");

        try {
            JasperFillManager.fillReportToFile(
                sourceFileName, parameters, beanColDataSource);
        } catch (JRException e) {
            e.printStackTrace();
        }
    }
}

```

The contents of the POJO file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBean.java** are as below:

```

package com.tutorialspoint;

public class DataBean {
    private String name;
    private String country;

    public String getName() {

```

```

        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }
}

```

The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBeanList.java** are as below:

```

package com.tutorialspoint;

import java.util.ArrayList;

public class DataBeanList {
    public ArrayList<DataBean> getDataBeanList() {
        ArrayList<DataBean> dataBeanList = new ArrayList<DataBean>();

        dataBeanList.add(produce("Manisha", "India"));
        dataBeanList.add(produce("Dennis Ritchie", "USA"));
        dataBeanList.add(produce("V.Anand", "India"));
        dataBeanList.add(produce("Shrinath", "California"));

        return dataBeanList;
    }

    /**
     * This method returns a DataBean object,
     * with name and country set in it.
     */
    private DataBean produce(String name, String country) {
        DataBean dataBean = new DataBean();
        dataBean.setName(name);
        dataBean.setCountry(country);
        return dataBean;
    }
}

```

Report Generation

We will compile and execute the above file using our regular ANT build process. The contents of the file build.xml (saved under directory C:\tools\jasperreports-5.0.1\test) are as below.

The import file - baseBuild.xml is picked from chapter [Environment Setup](#) and should be placed in the same directory as the build.xml.

```

<?xml version="1.0" encoding="UTF-8"?>
<project name="JasperReportTest" default="viewFillReport" basedir=".">
    <import file="baseBuild.xml" />
    <target name="viewFillReport"
        depends="compile,compilereportdesing,run"
        description="Launches the report viewer to preview
the report stored in the .JRprint file.">
        <java classname="net.sf.jasperreports.view.JasperViewer"
            fork="true">
            <arg value="-F${file.name}.JRprint" />
            <classpath ref />

```

```

    </java>
</target>
<target name="compilereportdesing"
    description="Compiles the JXML file and
    produces the .jasper file.">
    <taskdef name="jrc"
        classname="net.sf.jasperreports.ant.JRAntCompileTask">
        <classpath ref />
    </taskdef>
    <jrc destdir=".">
        <src>
        <fileset dir=".">
            <include name="*.jrxml" />
        </fileset>
        </src>
        <classpath ref />
    </jrc>
</target>
</project>

```

Next, let's open command line window and go to the directory where build.xml is placed. Finally execute the command **ant -Dmain-class=com.tutorialspoint.JasperReportFill** (viewFullReport is the default target) as follows:

```

C:\tools\jasperreports-5.0.1\test>ant -Dmain-class=com.tutorialspoint.JasperReportFill
Buildfile: C:\tools\jasperreports-5.0.1\test\build.xml

clean-sample:
[delete] Deleting directory C:\tools\jasperreports-5.0.1\test\classes
[delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jasper
[delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrprint

compile:
[mkdir] Created dir: C:\tools\jasperreports-5.0.1\test\classes
[javac] C:\tools\jasperreports-5.0.1\test\baseBuild.xml:28: warning:
'incldeantruntime' was not set, defaulting to build.sysclasspath=last;
set to false for repeatable builds
[javac] Compiling 3 source files to C:\tools\jasperreports-5.0.1\test\classes

compilereportdesing:
[jrc] Compiling 1 report design files.
[jrc] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
[jrc] log4j:WARN Please initialize the log4j system properly.
[jrc] log4j:WARN See
http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
[jrc] File : C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrxml ... OK.

run:
[echo] Runnin class : com.tutorialspoint.JasperReportFill
[java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
[java] log4j:WARN Please initialize the log4j system properly.

viewFillReport:
[java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
[java] log4j:WARN Please initialize the log4j system properly.

```

As a result of above compilation, a JasperViewer window opens up as in the screen below:



Here we can see that the color of the font is changed to blue for even count (in column country). In the column name, the font color is changed to green (this style is referenced from the style template).